## Lab 2.3 – Building and Testing MLB leaning models

In this project, the goal was to predict whether a player would win a Gold Glove based on their fielding statistics, using both a Decision Tree Classifier and a Random Forest Classifier. To start, the dataset was split into training and test sets based on the Training\_Validation column. Irrelevant columns like awardID, playerID, yearID, POS, and lgID were dropped to focus solely on the fielding statistics that would serve as features for the classification models.

For model training, both the Decision Tree and Random Forest Classifiers were tuned using GridSearchCV to optimize hyperparameters. The parameter grid included max\_depth, min\_samples\_split, min\_samples\_leaf, and class\_weight, which helped in finding the best-performing configuration for each model. Among the two classifiers, the Random Forest performed slightly better than the Decision Tree. It achieved a best score of 0.99067 compared to the Decision Tree’s score of 0.99056. The optimal model was configured with min\_samples\_leaf=5, which means that each leaf node in the tree must have at least 5 samples to prevent the model from overfitting by creating overly specific patterns. The min\_samples\_split=5 was used, ensuring that a node must have at least 5 samples to be considered for splitting, which helped maintain generalization. Additionally, the model used 10 trees (n\_estimators=10), striking a balance between model complexity and computational efficiency. After applying the models to the test set, the Random Forest Classifier showed high overall accuracy at 99%. However, the model had a significant issue with specificity—0.0, meaning it struggled with false positives. Specifically, the model had perfect sensitivity (1.0) but failed to accurately predict when a player would not win a Gold Glove, which suggests overfitting and poor handling of class imbalance.

A confusion matrix was used to assess the performance in more detail, revealing that while the model was great at identifying actual Gold Glove winners, it lacked in distinguishing players who wouldn't win. The ROC AUC score for the "No Gold Glove" class was 0.918, indicating room for improvement. This challenge with specificity suggests that rebalancing the classes or adjusting the class\_weight hyperparameter might help correct the imbalance, allowing the model to better handle cases where a player does not win.

In a separate regression task, the objective was to predict player salaries using batting, pitching, and fielding statistics, applying both a Decision Tree Regressor and a Random Forest Regressor. Similar to the Gold Glove prediction task, the dataset was split into training and test sets, and irrelevant columns like salary, playerID, and yearID were removed. GridSearchCV was used again to tune hyperparameters such as max\_depth, min\_samples\_split, min\_samples\_leaf, and for Random Forest, the number of estimators. The scoring metrics used included R² and negative mean squared error (MSE), to assess how well the models predicted salary.

In this regression task, the Random Forest Regressor outperformed the Decision Tree Regressor. The Random Forest achieved an R² score of 0.281, while the Decision Tree reached only 0.188. The best-performing Random Forest model had 100 estimators, no maximum depth, and a minimum of 5 samples per leaf. On the test set, the Random Forest Regressor showed an R² of 0.283, MSE of 25.88 trillion, and mean absolute error (MAE) of 3.27 million. Despite the Random Forest’s relative success, the low R² scores indicate that neither model captured the variance in player salaries well, likely due to the complexity and wide range of salaries in the dataset.

In conclusion, for the Gold Glove classification task, both models demonstrated high predictive accuracy, but the Random Forest performed better overall. However, the models struggled with specificity, indicating the need for rebalancing classes or tuning additional hyperparameters like class\_weight. As for the salary prediction regression task, the Random Forest Regressor again performed better than the Decision Tree, but the low R² scores suggest that more complex models or additional features may be necessary to improve predictions. Both tasks and models could be improved by eliminating problems of overfitting.